02 | 先利其器：如何高效学习 eBPF？

2022-01-19 倪朋飞

《eBPF 核心技术与实战》 课程介绍 >

讲述：山荣

时长 09:44 大小 8.92M

你好，我是倪朋飞。

上一讲，我们一起了解了 eBPF 的发展历程、基本原理和主要应用场景。eBPF 来源于 Linux 内核子系统 Berkeley Packet Filter (BPF)，最早用于提升网络包过滤的性能。后来，随着 BPF 技术的逐步完善，它的应用范围从内核空间扩展到了用户空间，并逐步在网络、可观测以及安全等方面获得了大量的应用。

了解过这些的你，很可能遇到了我曾经有过的疑惑：作为 Linux 内核的一部分，eBPF 这么底层的技术，到底该如何学习才能更高效地掌握它？

这是初学者经常遇到的问题：在学习 eBPF 的知识和原理时，找不到正确的方法，只是照着网络上并不全面的片段文章操作，或者直接去啃内核的源码，这样往往事倍功半。甚

至，还可能被海量的信息淹没，失去了持续学习的信心，“从入门到放弃” 。那么今天，我们就一起来看看，怎么才能高效且深入地学习 eBPF。

学习这门课需要什么基础？

首先，在学习 eBPF 之前你要明白，eBPF 是 Linux 的一部分，它所有的应用都需要在 Linux 系统中完成（虽然 Windows 也已经支持了 eBPF，但暂时不够成熟）。所以，我希望你**至少熟练掌握一种 Linux 系统（比如 Ubuntu、RHEL）的基本使用方法**，包括：

- 熟悉 Linux 系统常用命令的使用方法；
- 熟悉 Linux 系统常用软件包的安装方法；
- 了解 Linux 系统常用的网络工具和基本的网络排错方法。

因为这门课的核心在于 eBPF，所以在后续的内容中，我不会详细介绍这些基本的 Linux 系统使用方法。所以，你需要提前掌握这些基本知识。这样，接下来我在讲解 eBPF 时，你就更容易理解背后的工作原理和详细使用方法，也更容易明白每一步实践操作的具体含义。

其次，由于 eBPF 是内核的一部分，在实际使用 eBPF 时，大都会涉及一些内核接口的编程。所以，我希望你**有一定的编程基础（比如 C 语言和 Python 语言），并了解如何从源代码编译和运行 C 程序。**以 C 语言和 Python 语言为例，主要包括：

- 了解 C 语言和 Python 语言的基本语法格式，并能够看懂简单的 C 语言和 Python 语言程序；
- 了解 C 语言程序的编译方法，并了解 make、gcc 等常用编译工具的基本使用方法；
- 了解 C 语言和 Python 语言程序的调试方法，并能够借助日志、调试器等，在程序出错时排查问题的原因。

虽然这些内容看起来可能会很难，并且内容很多，但实际上任何一本 C 语言和 Python 语言的入门书籍都会讲到这些基本知识，这里我推荐适合编程入门者的 《C 程序设计语言》和《Python 编程：从入门到实践》（如果你更喜欢用视频的方式学习，也可以在 B 站中找到很多入门的视频教程）。

了解了 C 语言和 Python 语言的编程基础，我在后续讲解 eBPF 和 Linux 内核相关的编程接口时，你就能够更好地理解这些编程接口的原理，也更容易通过编程把 eBPF 变成自己的武器。

最后，虽然 eBPF 是 Linux 内核的一部分，其编程接口也通常涉及一定的内核原理，但**在刚开始学习时，我并不建议你深入去学习内核的源码**。就像我在《Linux 性能优化实战》专栏中提到的那样：学习一门新的技术时，你并不需要了解所有与这门技术相关的细节，只要了解这个技术的基本原理，并建立起整个技术的全局观，你就可以掌握这个技术的核心。

学习 eBPF，重点是什么？

在开篇词里，我已经提过这一点：学习一门新技术，最快的方法就是理解原理的同时配合大量的实践，eBPF 也不例外。因而，我们学习 eBPF 的重中之重就是：

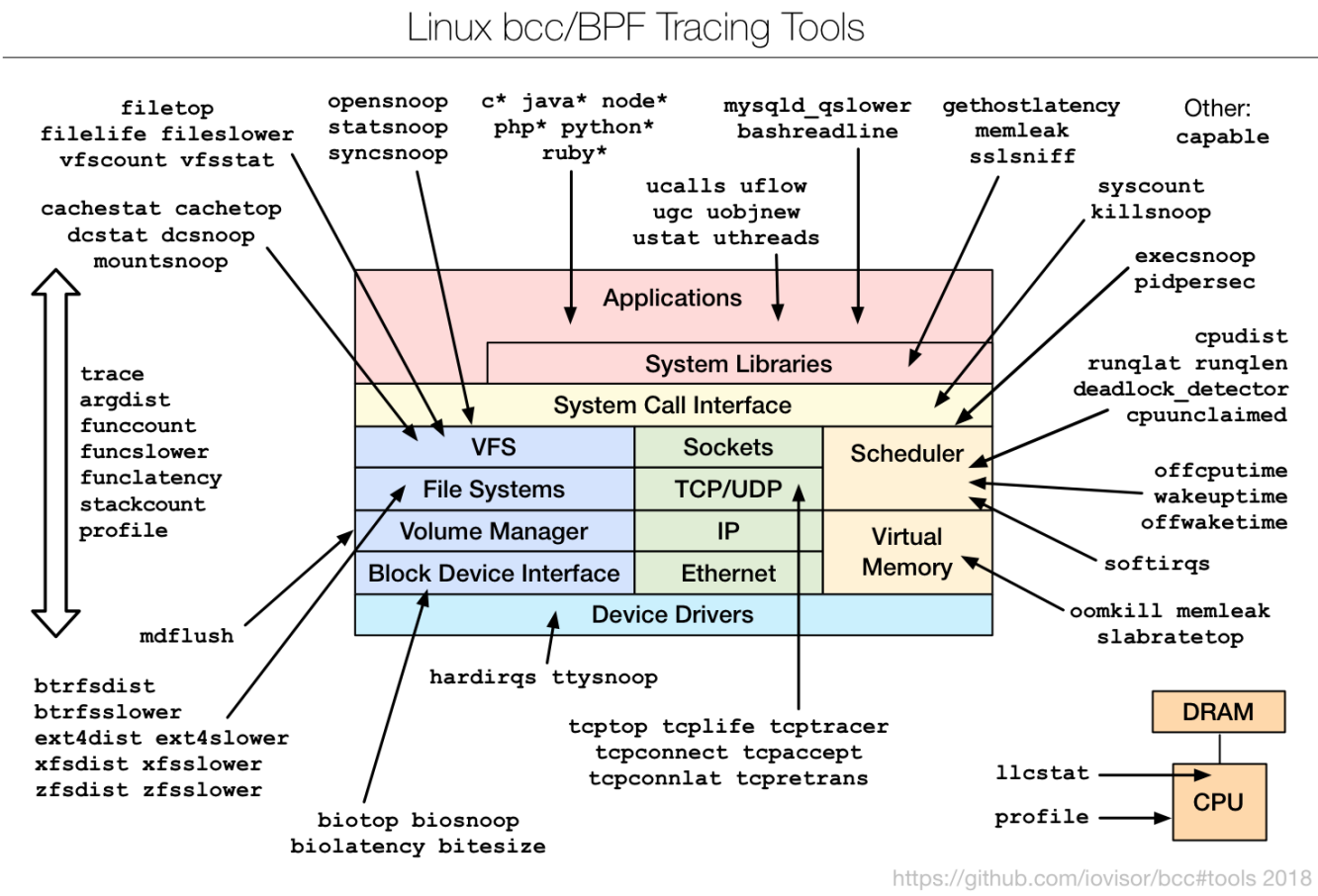
理解 eBPF 的基本原理；

掌握 eBPF 的编程接口；

通过实践把 eBPF 应用到真正的工作场景中。

这门课会针对不同场景，把这三个重点方面给你讲清楚，也希望你一定要花时间和心思来消化它们。

其实，如果你之前学习过我的上一个专栏《Linux 性能优化实战》，你就已经把 eBPF 用到了性能优化的场景中。作为最灵活的动态追踪方法，BCC 包含的所有工具都是基于 eBPF 开发的。如下图（图片来自 www.brendangregg.com）所示，BCC 提供了贯穿内核各个子系统的动态追踪工具：

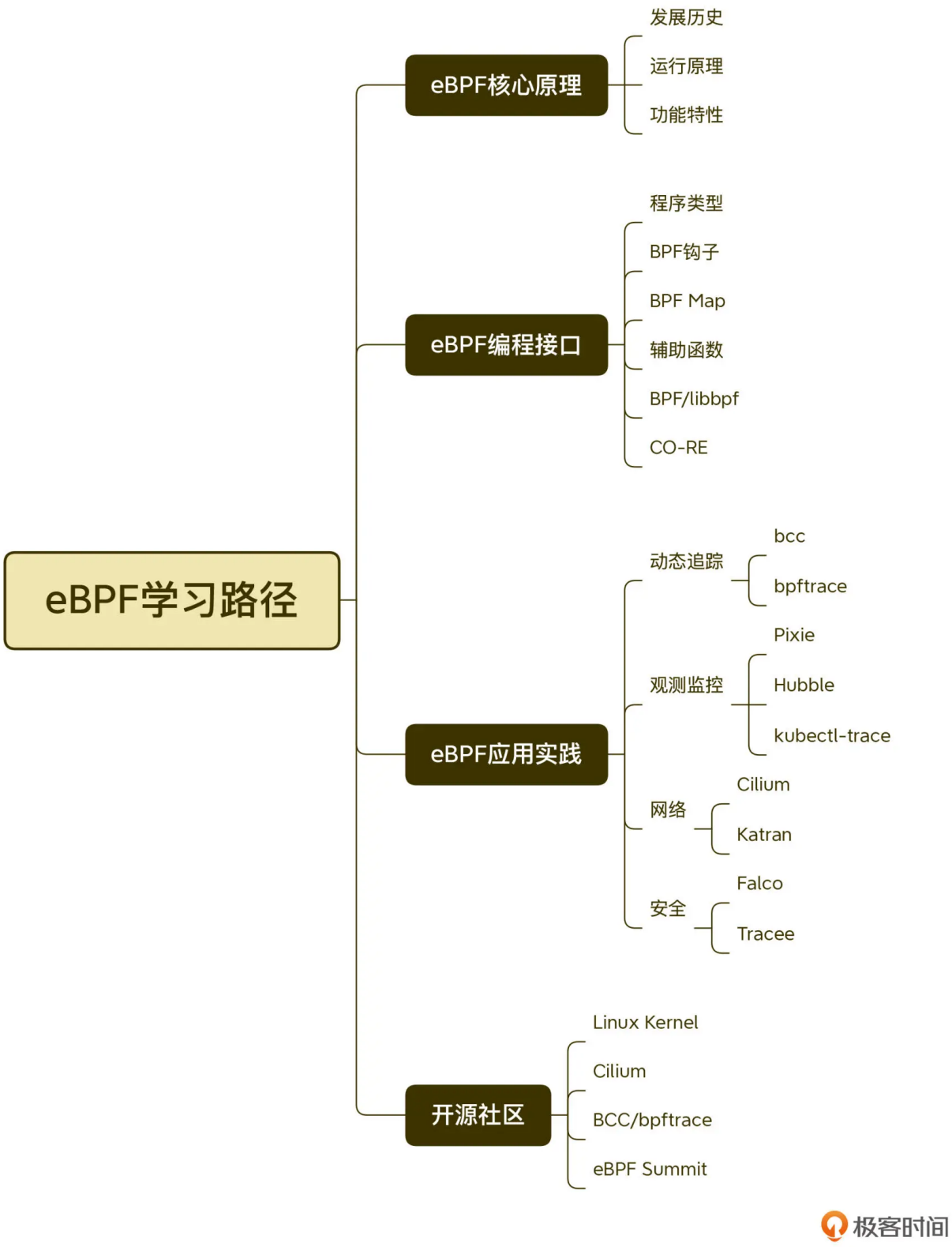


BCC 工具大全

所有的这些工具都是开源的，它们其实也是学习 eBPF 最好的参考资料。**你可以先把这些工具用起来，然后通过源码了解它们的工作原理，最后再尝试扩展这些工具，增加自己的想法。**

但是切记，BCC 等工具并不是学习的全部。工具只是解决问题的手段，关键还是背后的工作原理。只有掌握了 eBPF 的核心原理，并且结合具体实践融会贯通，你才能真正掌握它们。

最后，为了让你对 eBPF 有个全面的认识，我画了一张思维导图，里面涵盖了 eBPF 的所有学习重点，这门课中也基本都会讲到。你可以把这张图保存或者打印下来，每学会一部分，就在上面做标记，这样就能更好地记录并把握自己的学习进度。



怎么学习更高效？

前面，我给你讲了 eBPF 的学习重点，接下来我再跟你分享几个学习技巧。掌握了这些技巧，你可以学习得更轻松。

技巧一：虽然对内核源码的理解很重要，但切记，不要陷入内核的实现细节中。

eBPF 的学习和使用绕不开对内核源码的理解和应用，但并不是说学习 eBPF 就一定要掌握所有的内核源码。实际上，如果你一开始就陷入了内核的具体实现细节中，很可能会因为巨大的困难而产生退意。

你要记得，学习 eBPF 的目的并不是掌握内核，而是利用 eBPF 这个强大的武器去解决网络、观测、排错以及安全等实际的问题。因而，对初学者来说，掌握 eBPF 的基本原理以及编程接口才是学习的重点。最后再强调一遍，学习时要分清主次，不要陷入内核的实现细节中。

技巧二：边学习边实践，并大胆借鉴开源项目。

eBPF 的基本原理当然重要，但只懂原理而不动手实战，并不能真正掌握 eBPF。只有用 eBPF 解决了实际的问题，eBPF 才真正算是你的武器。

得益于开源软件，无论在 Linux 内核中还是在 GitHub 等开源网站上，你都可以找到大量基于 eBPF 的开源案例。在动手实践 eBPF 时，你可以大胆借鉴它们的思路，学习开源社区中解决同类问题的办法。

技巧三：多交流、多思考，并参与开源社区讨论。

作为一个实践性很强，并且在飞速发展的技术，eBPF 的功能特性还在快速迭代中。同时，由于新技术总有一个成熟的过程，在开始接触和应用 eBPF 时，我们也需要留意它在不同内核版本中的限制。

所以，我希望你在学习 eBPF 的过程中，做到多交流、多思考。你可以随时在评论区给我留言，写下自己的疑问、思考和总结，和我还有其他的学习者一起讨论切磋。看到优秀的开源项目时，你也可以去参与贡献，和开源社区的大拿们共同完善 eBPF 生态。

小结

今天，我带你一起梳理了高效学习 eBPF 的方法，并分享了一些我在学习 eBPF 时整理的技巧。

相对于其他技术来说，eBPF 是一个更接近 Linux 底层的技术。虽然学习它之前不需要你掌握内核的开发，但由于 eBPF 的应用总是伴随着大量的开发过程，你最好可以先熟悉 Linux

系统的基本使用方法，以及 C、Python 等语言的基础编程。

在学习 eBPF 技术时，你也不需要掌握所有相关技术的细节，只要重点把握 eBPF 基本原理和编程接口，再配合大量的工作实践，你就可以完全掌握 eBPF 技术。

当然了，eBPF 还处在飞速发展的过程中，多了解开源社区的进展和最新应用案例，多参与开源社区讨论，可以帮你更深刻地理解 eBPF 的核心原理，同时更好地把握 eBPF 未来的发展方向。

思考题

这一讲和上一讲的内容，都是为我们后续的学习做准备的，你可以把它们当作课前的热身环节。从下一讲开始，我们就要正式进入 eBPF 的世界了。所以，我想请你聊一聊：

1. 你之前有没有了解过基于 eBPF 的开源项目，可以和同学们分享你觉得优秀的项目吗？
2. 在学习和使用 eBPF 时，你有哪些心得，又遇到过哪些问题？欢迎在评论区分享。

今天的内容就结束了，欢迎把它分享给你的同事和朋友，我们下一讲见。

分享给需要的人，Ta订阅超级会员，你将得 50 元

Ta单独订阅本课程，你将得 20 元

生成海报并分享

赞 6 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 01 | 技术概览：eBPF 的发展历程及工作原理

下一篇 03 | 初窥门径：开发并运行你的第一个 eBPF 程序

精选留言 (3)

莫名

2022-01-19

写留言

曾基于 BPF 做过一个容器平台的链路追踪系统，分解出单个请求在服务端经过的节点、网络设备、耗时等信息，便于快速定位网络抖动时主要延迟的具体发生点。

遇到最多的是内核版本差异引起的各类编译问题，要么跑不起来，要么运行结果不符合预期。尤其 4.9 内核问题很多，5.x 版本的内核自己在测试环境用一用还行，线上的内核版...

展开

作者回复: 很赞的分享，谢谢！欢迎分享更多的实践经验。

6

ermaot

2022-01-19

从倪老师的linux性能篇，了解到了ebpf，就买了《bpf之巅》自学了一阵，现在居然倪老师也出了ebpf的课程，果断入手，希望认识能更上一个台阶

作者回复: 嗯嗯，这本书不错，我们一起加油！

共 3 条评论 >

5

includestdio.h

2022-01-19

第一次接触bpf是通过老师的 linux性能优化专栏，然后看到老师有推荐性能之巅这本书，果断入手并断断续续看完了，目前实际工作中还没有接触过ebpf，因此也无从入手，希望通过专栏能收获更多

作者回复: 感谢对专栏的支持，其实我们性能优化专栏里面已经用了很多的ebpf工具，这门课之后我们就可以自己按需来构建自己的性能优化工具了

共 2 条评论 >

1